Journal of Real-Time Image Processing (2021) 18:511–524
https://doi.org/10.1007/s11554-020-00983-y

ORIGINAL RESEARCH PAPER

Complexity reduction of test zonal search for fast motion estimation in uni-prediction of High Efficiency Video Coding

K. C. Ravi Chandra Varma[1] · Sudipta Mahapatra[1]

Received: 25 October 2018 / Accepted: 8 May 2020 / Published online: 8 June 2020

Abstract

The complexity of motion estimation in the High Efficiency Video Coding (HEVC) standard is very high as it uses a number of prediction block sizes. The test zonal search (TZS) mechanism is used as motion estimation algorithm in the fast search mode of HM encoder, the reference software for HEVC. In this paper, we present schemes for reducing the complexity of the TZS algorithm for uni-prediction in HEVC. The proposed mechanisms help in reducing complexity of grid, raster and refinement search stages of TZS. The performance of the proposed mechanisms is tested independently and also combined in the fast search mode of HM-16.18. The motion estimation time and the number of search points in the combined algorithm are reduced by 68.14% and 77.10% with a BD-rate of 0.30% and a BD-PSNR of −0.007 dB in comparison with the original fast search mode in the low-delay P main profile. The proposed complexity reduction schemes are compared with those adopted in other recently proposed fast motion estimation algorithms and the performance of the proposed scheme is found to be superior in terms of the reduction in motion estimation time.

Keywords Motion estimation · Adaptive search range · Motion estimation termination · High Efficiency Video Coding

1 Introduction

Joint Collaborative Team on Video Coding (JCT-VC) has developed High Efficiency video Coding (HEVC) with the goal of improving the compression ratio [1]. As compared to H.264 [2], its earlier standard, HEVC provides nearly 50% bit rate reduction, albeit at the cost of a high computational complexity due to the adoption of advanced video coding tools like advanced motion vector prediction (AMVP), large-sized coding tree units (CTUs), recursive quad-tree structure partitioning of CTU, use of an interpolation filter for fractional pixel motion estimation and variable transform sizes. There is a need for reducing the complexity of HEVC without significantly affecting the video quality and amount of compression.

The motion estimation (ME) block of HEVC is comparatively more complex than the other components. ME is carried out for different block sizes with the individual blocks organized in a recursive quad-tree structure. Though full search motion estimation provides an optimal solution, its complexity is very high contributing to around 80–90% of the total encoding complexity. In the fast search mode of the HM encoder, the test zonal (TZ) search algorithm [3] is employed to reduce the motion estimation time without any significant effect on bit rate and video quality [4].

The TZ search algorithm includes four stages, namely motion vector prediction, grid search, raster search and refinement search stages. In the HM encoder, the various stages are implemented as follows:

Motion vector prediction In this stage among zero MV predictor, AMVP and upper depth predictor, the best predicted motion vector (PMV) is found based on the minimum rate-distortion (RD) cost. Upper depth predictor is not used in the older versions of the HM encoder.

In Fig. 1, the prediction motion vector PMV2 is a better predicted motion vector as compared to the predicted motion vector PMV1. Because PMV2 is same as MV, the actual motion vector, when RD cost calculation is done with diamond pattern with stride distance $s = 1, 2$ and 4, the minimum cost search point will not change for three consecutive strides. Hence, grid search can be terminated. In grid search

✉ K. C. Ravi Chandra Varma
kcravi912@gmail.com

Sudipta Mahapatra
sudipta@ece.iitkgp.ac.in

[1] Department of E & ECE, IIT Kharagpur, Kharagpur, West Bengal 721302, India

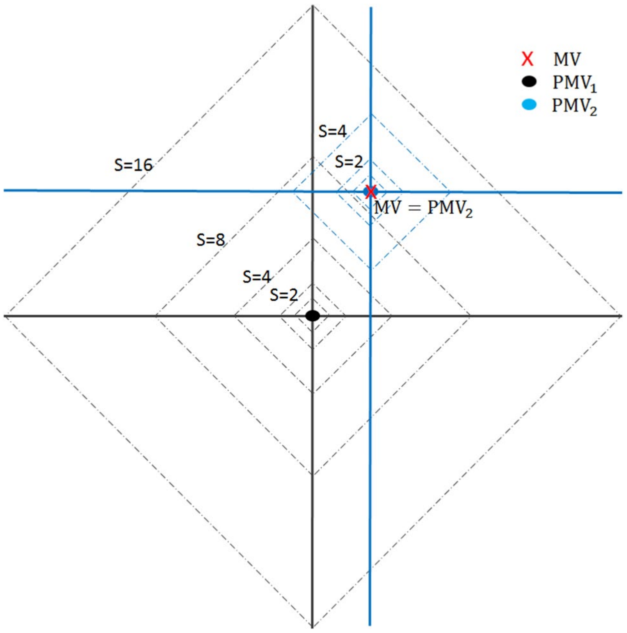

Fig. 1 Grid search with each search point numbered

for PMV1, the MV is near to its diamond pattern with $s = 8$. So, cost calculation continues till $s = 64$.

Grid search stage The PMV position obtained from motion vector prediction is used as the search centre in the grid search stage. Diamond search patterns are used by default in this stage with stride distances of 1, 2, 4 and so on in powers of 2 up to the maximum search range. These patterns are used to find the search point corresponding to the minimum RD cost. In the HM encoder, for diamond search patterns, the number of search points (n) for different stride distances (s) is as given in (1). For a search range of 64, there are at most $4 + 3 \times 8 + 3 \times 16 = 76$ search points. In this stage if the minimum RD cost is not changed for three consecutive strides, then grid search is terminated

$$n = \begin{cases} 4, & s = 1 \\ 8, & 2 \leq s \leq 8 \\ 16, & 8 < s \end{cases} \tag{1}$$

In Fig. 1, if the minimum cost search point does not change for three consecutive strides, it is highly likely that the actual motion vector is near to this search point. In the TZ search of HM encoder, grid search is terminated if the minimum cost search point does not change three consecutive strides, thereby optimizing the grid search in terms of complexity.

Raster search stage Raster search is used only if in grid search the distance of the minimum cost search point from the search centre is greater than 5. Raster search is similar to full search with the search window downsampled by a raster distance both horizontally and vertically. In the HM encoder, the raster distance is set to 5 by default. With the search range of 64, the search area is of size $(2 \times 64 + 1) \times (2 \times 64 + 1) = 129 \times 129$. Therefore, the search area has a total of $\text{ceil}\left(\frac{129}{5}\right) \times \text{ceil}\left(\frac{129}{5}\right) = 26 \times 26 = 676$ search points.

Refinement search stage Star refinement or raster refinement search is used in the refinement search stage of the TZ search algorithm. In HM encoder, star refinement is enabled by default. Initially, the refinement search centre is the search point corresponding to the minimum RD cost obtained in the previous stages. Starting from the refinement search centre, the search process is carried out with diamond pattern strides as in grid search. After obtaining the minimum cost search point, refinement search is terminated if the this search point is same as the refinement search centre. If the distance of the minimum cost search point from the refinement search centre is one, then a two-point search is carried out to check whether the minimum cost search point changes. Refinement search is terminated if there is no change after two-point search. If the distance of the minimum cost search point from the refinement search centre is greater than 1 or if the distance is 1 and the minimum cost search point changes with two-point search, then refinement search is again performed with the new refinement search centre at the minimum cost search point. Refinement search is repeated till the minimum cost search point is at a distance of 0 from the refinement search centre or with a distance of 1 with the additional condition that the minimum cost search point does not change with two-point search.

Table 1 shows the percentage of the number of search points ($\%N$) and complexity in terms of the number of 4×4 blocks ($\%C$) in the grid, raster and refinement search stages. In this table, the ’Others’ column includes the complexity involved in calculating the best PMV among AMVP, zero motion vector predictor and the upper depth predictor. The number of search points in the raster search stage is large compared to the grid search stage. For a bigger search range, there is a huge difference in number of search points between these two stages. Though the complexity involved in the raster search stage is much higher than the grid search and refinement search stages, the number of search points and the number of 4×4 blocks for a sequence in raster search are comparable to that of grid search. This is because raster search is enabled for prediction blocks (PBs) belonging to regions having high motion and multi-directional motion. For sequences like Kimono, Cactus, BasketballDrive, BasketballDrill, BQMall, RaceHorsesC, BasketballPass and RaceHorses, the number of 4×4 blocks is larger for raster search compared to grid search. The BQSquare sequence has a lower number of

Table 1 Percentage complexity distribution of TZ search algorithm of HM-16.18 encoder in LD-P main profile

Class	Sequence	%*N*				%*C*			
		Grid	Raster	Refinement	Others	Grid	Raster	Refinement	Others
Class B	Kimono	33.9	42.5	20.0	3.6	24.3	52.7	20.7	2.3
	ParkScene	56.0	24.0	13.4	6.5	42.6	35.2	17.6	4.7
	Cactus	45.4	36.2	14.8	3.7	33.0	48.0	16.5	2.5
	BasketballDrive	31.4	47.3	18.0	3.3	21.0	58.3	18.8	1.9
	BQTerrace	62.3	18.7	11.8	7.2	49.2	30.2	15.1	5.5
Class C	BasketballDrill	40.6	40.1	16.0	3.3	29.6	51.4	16.8	2.2
	BQMall	40.9	38.7	17.0	3.5	30.1	49.4	18.2	2.4
	PartyScene	43.7	36.7	15.4	4.2	35.9	44.8	16.0	3.3
	RaceHorsesC	28.4	49.2	19.5	3.0	20.6	58.2	19.4	1.8
Class D	BasketballPass	43.0	36.1	16.9	4.0	31.2	47.3	18.8	2.7
	BQSquare	81.8	4.5	5.8	7.9	80.6	4.8	6.8	7.9
	BlowingBubbles	49.3	28.6	17.0	5.1	42.1	34.7	19.0	4.2
	RaceHorses	31.4	44.7	20.6	3.3	24.3	52.5	20.9	2.3
	Average	45.2	34.4	15.9	4.5	35.7	43.7	17.3	3.4

4 × 4 blocks for raster search compared to grid search. Therefore, it is necessary to reduce the number of search points in both grid search and raster search stages.

This paper proposes four different schemes to reduce the ME complexity for the TZ search algorithm in uni-prediction. The proposed methodologies include an adaptive search range algorithm and schemes to reduce the complexity of grid search, raster search and star refinement search. The rest of the paper is organized as follows. In Sect. 2, recently proposed fast motion estimation algorithms are described in brief. Section 3 presents schemes for reducing the motion estimation complexity in uni-prediction. Experimental results are presented in Sect. 4, and the paper concludes in Sect. 5.

TZ search can be used for fast motion estimation in uni-prediction of HEVC. In bi-prediction of HM encoder, full search is used within a search range of 4. The proposed schemes are limited to TZ search, which is used for fast motion estimation in the HM encoder. The TZ search algorithm gives a better trade-off between complexity reduction (compared to full search) and the bit rate and PSNR (not a significant change in bit rate and PSNR compared to full search). There is no significant difference in performance of TZ search and full search motion estimation algorithms in terms of bit rate and PSNR. But TZ search has huge reduction in complexity compared to full search.

2 Related work

2.1 Fast search pattern-based motion estimation

Fast search pattern-based motion estimation algorithms like 2D-logarithmic search [5], cross search [6], three-step search [7], four-step search [8], diamond search [9] and hexagonal search [10] lead to significant reduction in the motion estimation complexity. However, when these fast search patterns are used for video sequences having high motion activity, the quality and the amount of compression are reduced due to the local minimum problem of these fast search patterns.

2.2 Adaptive search range algorithm

To find the best match reference block for the current macroblock, the search operation is carried out over all the reference macroblocks in the search window. Many adaptive search range algorithms reduce the motion estimation complexity by reducing the size of the search window based on the neighbouring information [11–17]. [11] also uses the sum of absolute difference (SAD) at the search centre to estimate the search range with the aim of reducing the motion estimation complexity for the full search algorithm. The linear adaptive search range in [12] is able to reduce the search range in uni-prediction with a minor increment in (Bjontegaard delta bitrate) BD-rate of 0.7% on an average. The scheme proposed by Nalluri et al. [13] uses the motion vector difference (MVD) of the spatial, temporal and upper depth neighbours for setting the search range. The algorithm presented in [14] is based on neighbouring MVDs modelled using a Laplace distribution. The mechanism proposed in [15] is based on neighbouring MVDs and models the MVDs of all the macroblocks in the previous frame with a Cauchy distribution. Cauchy distribution can be used to model the global search range in terms of MVDs of the previous frame. The local search range is based on the global search range and the MVDs of spatial and collocated neighbouring blocks as given in [15].

[16] uses the depth information for adapting the search range. Our previous work [17] generates a better prediction of search range by using MVDs of the neighbouring blocks, modelled using a Cauchy distribution and the SAD at the search centre.

2.3 Motion estimation termination

The early stop search algorithm adopted in [11] and [18] terminates the ME process if the current search point in the spiral search pattern has the minimum RD cost. The termination criterion is based on a threshold obtained from the minimum RD cost among the costs of neighbouring blocks. Termination of the grid search stage is done in the HM encoder if the diamond patterns in the previous two strides and the current stride do not have the minimum RD cost.

2.4 Partial distortion search

Partial distortion search (PDS)-based mechanisms reduce the motion estimation complexity by using SADs of subsampled blocks [19]. The normalized partial distortion search mechanism used in [20] leads to further reduction in the motion estimation complexity. The adjustable partial distortion search used in [21] can be used to vary degrees of motion estimation complexity with variations in the video quality. Block size-based sub-sampling factors are used in [22]. Additionally, in adjusted [22] thresholds over multiple reference frame are adjusted leading to further reduction in motion estimation complexity.

For reducing the motion estimation complexity of TZ search, in [23] grid search is replaced by rotating pentagon search, which reduces the motion estimation time to a certain extent. The search range is repeatedly reduced in [24] by using a quadratic function in order to get an optimal motion vector (MV). This method reduces the complexity with a slight loss of coding efficiency. The fast and low complexity motion estimation approach proposed in [25] for ultra-high-definition videos greatly reduces the motion estimation complexity compared to the full search algorithm at the expense of a minor degradation in coding efficiency. Search ranges based on video characteristics and the successive elimination algorithm are used in [26]. This method also leads to a minor degradation in the coding efficiency.

By classifying the current prediction block as one with smooth, medium or complex motion and employing a different motion estimation strategy for each class, [27] is able to reduce the ME complexity with negligible effect on the coding efficiency. [28] uses two AMVPs and the predicted motion vectors of blocks at a lower depth to reduce the complexity of bottom up integer motion estimation without significantly affecting the coding efficiency.

3 Motion estimation complexity reduction schemes for uni-prediction

In terms of compression and video quality, the TZ search algorithm, a highly optimized ME strategy, has a performance similar to that of full search, though it a much lower complexity.

3.1 Adaptive search range algorithm

Use of better predictors will make the search range prediction more accurate, providing a better trade-off between complexity, BD-rate and BD-PSNR. X and Y components of MVD are obtained as given in (2)

$$\begin{aligned}\mathrm{MVD}_X &= \mathrm{MV}_X - \mathrm{PMV}_X\\ \mathrm{MVD}_Y &= \mathrm{MV}_Y - \mathrm{PMV}_Y\end{aligned} \tag{2}$$

where MVD_X and MVD_Y are MVDs, MV_X and MV_Y are the motion vectors and PMV_X and PMV_Y are predicted motion vectors in x and y directions, respectively.

The distance of the MVD from the PMV, $d(X, Y)$, is obtained as given in (3)

$$d(X,Y) = \sqrt{\mathrm{MVD}_X^2 + \mathrm{MVD}_Y^2} \tag{3}$$

SR_1, the spatial search range predictor is obtained from the length of MVD which is the distance of the MV from the PMV as given in (4)

$$\mathrm{SR}_1 = k_1 \times max\left(d_\mathrm{L}, d_\mathrm{U}, d_\mathrm{UR}\right) \tag{4}$$

where d_L, d_U and d_UR are the length of MVDs of left, upper and upper right block. k_1 is a constant.

As specified in (5) the second search range predictor, SR_2 is obtained using the MVD distribution of the previous frame. The MVDs of a frame are represented accurately by a Cauchy distribution [15]. X and Y components of the predicted search range are denoted by C_X and C_Y

$$\begin{aligned}\mathrm{SR}_2 = max(&C_X + k_2 \times \mathrm{maxMVD}_X,\\ &C_Y + k_2 \times \mathrm{maxMVD}_Y)\end{aligned} \tag{5}$$

where k_2 is a constant. For each frame in RA mode, there are up to two neighbouring frames chosen for search range prediction. One is the immediate neighbouring frame in the forward direction, and the other is the immediate neighbouring frame in the backward direction that are already coded before the current frame. Each of C_X and C_Y is obtained as the maximum of the predicted search ranges of the forward and backward directions.

Current block's motion vector is proportional to the SAD corresponding to the PMV position ($\mathrm{SAD}_\mathrm{PMV}$). This means the high motion prediction block (PB) has high $\mathrm{SAD}_{\mathrm{PMV}}$ value, and the low motion PB has low $\mathrm{SAD}_{\mathrm{PMV}}$ value. In the beginning of TZ search, PMV is chosen from AMVP, zero motion and upper depth predictors based on the minimum RD cost. This PMV is used as the search centre for grid search.

The third search range predictor, SR_3, is obtained from the $\mathrm{SAD}_{\mathrm{PMV}}$ normalized by block size and is given in (6)

$$\mathrm{SR}_3 = k_3 \times \frac{\mathrm{SAD}_{\mathrm{PMV}}}{n_1 \times n_2} \tag{6}$$

where $n_1 \times n_2$ is the block size. Low-resolution sequences have smaller value of k_3 and high-resolution sequences have larger value.

In the proposed adaptive search range mechanism, a total of four search ranges are used. Three of the search ranges are the same as in our previous work [17]. SR_4, the fourth search range is derived from the deviation of the predicted motion vector (PMV) with respect to the motion vectors of the neighbouring blocks.

The fourth search range predictor is obtained from the predicted motion vector deviation (PMVD). MVD is the deviation of PMV from the actual MV. PMVD is defined as the maximum of the deviations of PMV from motion vectors of the neighbouring blocks. PMVD is used for predicting the search range. PMVD over x and y directions is calculated as given in (7). SR_4, the search range, is predicted as the maximum of PMVDs in x and y directions

$$\begin{aligned}\mathrm{PMVD}_X &= \max_{1\le i\le n}(\mathrm{PMV}_X - \mathrm{MV}_{X(i)})\\ \mathrm{PMVD}_Y &= \max_{1\le i\le n}(\mathrm{PMV}_Y - \mathrm{MV}_{Y(i)})\end{aligned} \tag{7}$$

where n is the number of neighbouring blocks considered for calculating the PMVD, (PMV_X, PMV_Y) is the PMV of the current block and ($\mathrm{MV}_{X(i)}$, $\mathrm{MV}_{Y(i)}$) represents MV of the ith neighbouring block

$$\mathrm{SR}_4 = k_4 \times max\left(\mathrm{PMVD}_X, \mathrm{PMVD}_Y\right) \tag{8}$$

We first define a search range, SR_I as given in (9). In this equation, $C_{0.996}$ is defined as the search range predicted from the Cauchy distribution with a probability of 0.996. The final search range is computed as specified in (10)

$$\mathrm{SR}_I = \begin{cases} max(\mathrm{SR}_1, \mathrm{SR}_2, \mathrm{SR}_3, \mathrm{SR}_4, 8 + C_{0.996}), n \le 2 \\ max(\mathrm{SR}_1, \mathrm{SR}_2, \mathrm{SR}_3, \mathrm{SR}_4), \text{otherwise} \end{cases} \tag{9}$$

$$\mathrm{SR} = max\left(\mathrm{SR}_I, 8 + min\left(a, max\left(C_X, C_Y\right)\right)\right) \tag{10}$$

In the above equations, the values of k_1, k_2, k_3, k_4 and a have an effect in motion estimation complexity and coding efficiency. The value of these parameters is adjusted in order to obtain a better trade-off between complexity and coding efficiency.

3.2 Motion estimation complexity reduction in the grid search stage

The complexity of the grid search stage is reduced by early termination of the ME process as well as by skipping half of the search points as discussed below.

Early termination of ME If the best minimum search point is not changed for one or two consecutive diamond pattern strides, then the grid search is terminated to reduce the grid search complexity. This is done if the number of neighbours is greater than 1. Let AMVP_1 and AMVP_2 be the motion vectors of the best two neighbouring blocks obtained from advanced motion vector prediction and J_{PMV} be the RD cost of the centre of the grid search stage. The grid search stage is terminated if the search point corresponding to the minimum RD cost is not changed for one diamond pattern stride, and if the following criteria are satisfied

$$\mathrm{AMVP}_1 = \mathrm{AMVP}_2 \tag{11}$$

$$J_{\mathrm{PMV}} < T \tag{12}$$

here, T, the threshold is fixed based on the size of the prediction block.

If the termination of grid search when minimum RD cost does not change for one stride is done all the time, then there would be a significant reduction in quality and compression. To take care of the quality and compression, grid search is terminated if the minimum RD cost is not changed after one stride when AMVP_1 and AMVP_2 are equal. This implies that the deviation between these two AMVPs is equal to 0. Even a small deviation between AMVP_1 and AMVP_2 can be used as it helps in terminating grid search more frequently, but it leads to slight loss in coding efficiency.

If the best minimum search point is not changed for two diamond pattern stride, grid search is terminated provided the following criterion is satisfied

$$max(\mathrm{SR}_1, \mathrm{SR}_4) = 0 \tag{13}$$

where SR_1 and SR_4 are the search ranges obtained from the spatial neighbours and PMV deviation. This implies that the grid search stage can be terminated quickly if these two search ranges are zero.

Half round in the grid search stage In the grid search stage, all the search points of diamond strides are used for the calculation of SAD and RD cost as shown in Fig. 2. Since the search point in the current round having the minimum RD cost is likely to be near the minimum cost search point of the previous round, only the search points near the

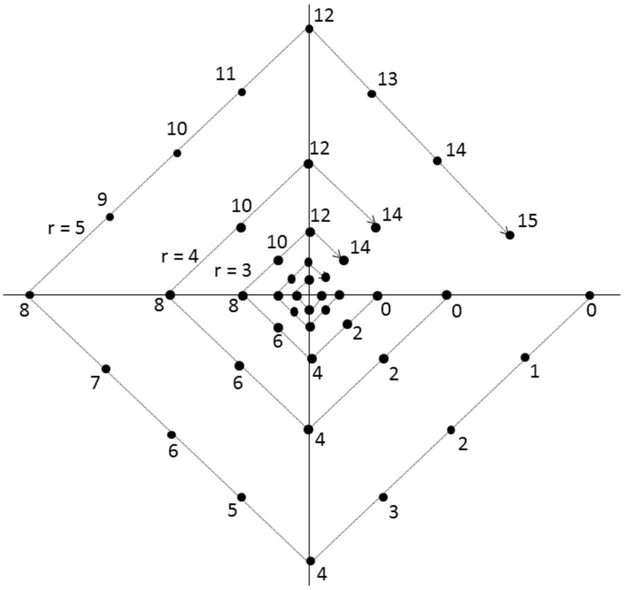

Fig. 2 Grid search with each search point numbered

minimum cost search point of the previous round are considered for the calculation of SAD and RD cost in this round.

For a diamond pattern stride having N search points, $(\frac{N}{2}+1)$ search points near the minimum RD cost search point of the previous round are used for calculating the SAD and RD cost. From Fig. 2, if the minimum RD cost is in round 4 at the search point number 8 and the current round, r is 5, then the search points numbered from 4 to 12 of round 5 are used for calculating the SAD and RD cost. Other search points in round 5 are skipped.

3.3 Motion estimation complexity reduction in the raster search stage

Search range reduction for raster search Raster search is used only if the search point in the grid search stage corresponding to the minimum RD cost falls beyond a distance of 5 from the search centre. Otherwise, it is not used. The use of raster search over the complete search range increases the complexity to a greater extent even if the best match search point is not much far away from the search centre. The search point having the minimum RD cost is likely to be near the search centre. If the minimum cost search point in the grid search stage is far from the search centre, the minimum cost search point obtained in the raster search stage is more likely to be far away from the minimum cost search point of the grid search stage. This implies that it is not required to perform raster search always over the entire search range. The search point having the minimum RD cost in the grid search is used as the search centre in the raster search stage. The search range for raster search is chosen selectively based on how far the minimum RD cost search point is from its search centre in the grid search stage as given in (14)

$$\mathrm{SR}_{\mathrm{raster}} = \begin{cases} 10, \ d_{\mathrm{grid}} <= 16 \\ 20, \ d_{\mathrm{grid}} <= 32 \\ 30, \ otherwise \end{cases} \tag{14}$$

In the grid search stage, two search points are chosen. If these two search points are nearby, then the search range is reduced for the raster search stage. If these two search points are far away from each other, then reducing the search range for raster search creates a local minimum problem. In order to determine whether the two minimum cost search points are nearby, search points in the diamond pattern stride are numbered as shown in Fig. 2. If the distance between the numbers allotted to these two search points is in between 4 and 12, then the minimum cost search point is considered to be in the direction of the minimum cost search point in the grid search stage. If these two best minimum search points are far away, i.e. the difference between the two search point numbers is greater than 4 and less than 12, then adaptive raster search is disabled and the search centre in the grid search stage is used as the search centre of raster search.

Two-level scanning order of raster search In Fig. 3a, b, the distance between two adjacent search points is 5. The SAD and RD cost are calculated at level 1 search points represented in dark colour in the order they are numbered. The level 1 search points are covered in a spiral scan manner in clockwise direction. We have chosen clockwise in our implementation. It does not make a difference in performance even if the patterns in grid, raster and refinement search are implemented anticlockwise. After calculating the RD cost at one search point in level 1, a threshold T_{L1} is computed as given in (15). In this equation, $J_{\min}$ is the minimum of all the previously calculated RD costs of level 1 and level 2 search points and PBsize, which is the size of the PB equals $n_1 \times n_2$. RD costs at the eight neighbouring search points represented in grey colour and connected by dotted lines are calculated if the criterion given in (16) is satisfied

$$T_{L1} = \begin{cases} J_{\min} + \mathrm{PBsize}, & J_{\min} < \mathrm{PBsize} \\ 2 \times J_{\min}, & J_{\min} < 6 \times \mathrm{PBsize} \\ J_{\min} + 6 \times \mathrm{PBsize}, & \text{otherwise} \end{cases} \tag{15}$$

$$J_{L1} < T_{L1} \tag{16}$$

where J_{L1} is the RD cost at the current level 1 search point.

As shown in Fig. 3a for $\mathrm{SR}_{\mathrm{raster}} = 10$, there are four level 1 search points. There are three level 2 search points corresponding to the two adjacent level 1 search points. And, the search centre for raster search is common for the four

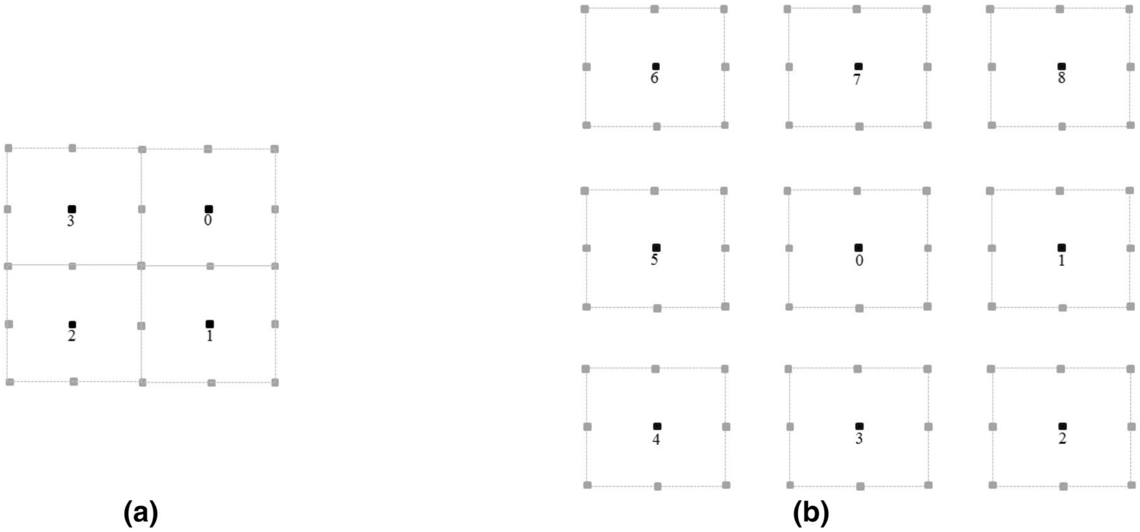

Fig. 3 Raster search represented in two-level scanning of search points with dark colour and grey colour search points representing level 1 and level 2, respectively, for **a** $\mathrm{SR}_{\mathrm{raster}} = 10$ and **b** $\mathrm{SR}_{\mathrm{raster}} = 20$

adjacent level 1 search points. If there are any level 2 search points corresponding to the previous level 1 search point and the current level 1 search point, and if those level 2 search points are already used for RD cost calculation, then these search points are not used again. For $\mathrm{SR}_{\mathrm{raster}} = 20$ as shown in Fig. 3b, there are no level 2 search points common to two adjacent level 1 search points. For $\mathrm{SR}_{\mathrm{raster}} > 20$, level 1 search points as given in Fig. 3b, which are considered in a spiral scan manner, can be extended over the search range and its corresponding level 2 search points are used for RD cost calculation only if the RD cost at level 1 satisfies the criterion given in (16).

3.4 Motion estimation complexity reduction in star refinement search

Adaptive search range in star refinement search In star refinement search stage of the TZS algorithm, diamond patterns with strides varying from 1 to 64 in multiples of 2 are used for finding the minimum RD cost. The minimum RD cost search point in this stage is more likely to be near the minimum RD cost search point obtained so far in the grid and raster search stages. Therefore, the search range can be reduced even at this stage as follows

$$SR_{refine} = \begin{cases} 2 \times d_{\mathrm{grid}}, & d_{\mathrm{grid}} <= 4 \\ 4, & \text{otherwise} \end{cases} \tag{17}$$

Redundant search point skip in refinement search If the best minimum is near the search centre in the grid search stage, then there are few search points in the refinement search stage that are already used for calculating the RD cost during grid search. This gives us the information that these search points do not correspond to the minimum RD cost. Therefore, RD cost calculation is skipped at these search points. In Fig. 4, the search points of the grid search stage are represented in black colour and grey colour within the diamond pattern strides represented in grey colour. In this figure, the minimum search point is at motion vector

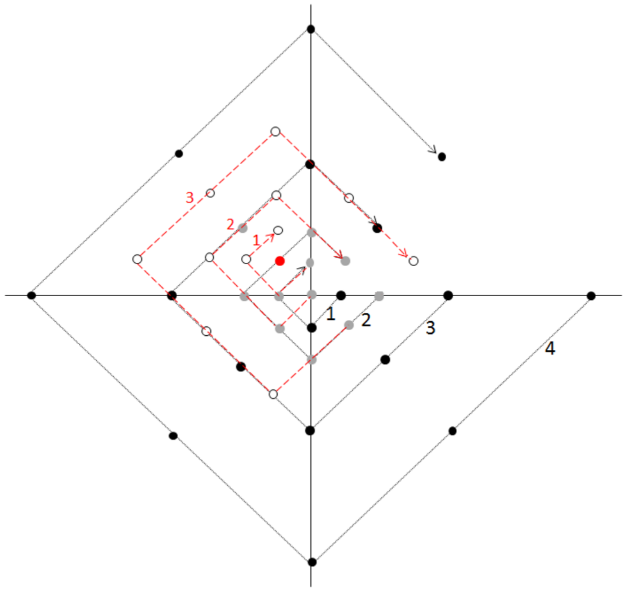

Fig. 4 Search points of diamond pattern strides of star refinement search represented in red lines and the diamond pattern strides of grid search represented in grey lines for best minimum in grid search at (−1, −1) represented in red shaded circle

difference of $(-1,-1)$, represented in red colour. The search points in the star refinement stage are represented in grey colour and blank circles within the diamond strides of red colour. Grey colour search points are common to both grid and star refinement search stages. Therefore, only search points represented by blank circles are used for RD cost calculation in the star refinement stage.

4 Experimental results

The proposed methods are implemented individually and then are combined in the HEVC reference software HM-16.18 [4] in the fast search mode. The schemes presented in [27] and [28] are implemented in HM-16.18 for fair comparison with the proposed scheme. The HM encoder is used in Visual studio 2013 in the release mode in an Intel Xeon 12-Core 2.5 GHz processor with 32-GB RAM. LD-P configuration uses only uni-prediction, and it is predicted from the previous frames only. Complexity is less in LD-P profile. LD-B configuration uses both uni-prediction and bi-prediction, and it is predicted from the previous frames only. Due to the use of bi-prediction, complexity increases although it provides better compression. The RA main configuration uses both uni-prediction and bi-prediction, and it is predicted from both previous and future frames in hierarchical structure. Due to the use of hierarchical structure, compression is high, but it has high complexity compared to other profiles if same number of reference frames are used in each reference list (same number even in LD-P and LD-B profile). Simulations are carried out in the LDP main configuration as specified in HEVC common test conditions [29]. Class B, C and D sequences obtained from [30] are used for evaluating the performance of the proposed and other existing algorithms. For each of the test sequences, the first hundred frames are used in the experimentation. In addition to Bjontegaard Delta PSNR (BD-PSNR) and Bjontegaard delta bitrate (BD-rate) [31], four other measures are used to evaluate the performance of the schemes. The additional measures are the percentage reduction in number of search points (ΔN), percentage reduction in the number of 4×4 blocks (ΔC), percentage reduction in motion estimation time (ΔMET) and the percentage reduction in encoding time (ΔET). These are calculated as follows:

$$\Delta N = \frac{N_{\mathrm{HM}} - N_{\mathrm{Proposed}}}{N_{\mathrm{HM}}} \times 100\% \tag{18}$$

$$\Delta C = \frac{C_{\mathrm{HM}} - C_{\mathrm{Proposed}}}{C_{\mathrm{HM}}} \times 100\% \tag{19}$$

$$\Delta \mathrm{MET} = \frac{\mathrm{MET}_{\mathrm{HM}} - \mathrm{MET}_{\mathrm{Proposed}}}{\mathrm{MET}_{\mathrm{HM}}} \times 100\% \tag{20}$$

$$\Delta \mathrm{ET} = \frac{\mathrm{ET}_{\mathrm{HM}} - \mathrm{ET}_{\mathrm{Proposed}}}{ET_{\mathrm{HM}}} \times 100\% \tag{21}$$

where the subscripts *HM* and *Proposed* denote the performance measures corresponding to the HM encoder and the proposed algorithm, respectively. Each of the measures $\mathrm{MET}_{\mathrm{Proposed}}$ and $\mathrm{ET}_{\mathrm{Proposed}}$ includes the overhead time due to the proposed scheme. It may be noted that the number of 4×4 blocks used for SAD calculation better reflects the complexity of an algorithm as compared to the number of search points covered. Because, for one search point in motion estimation, PBs of different sizes have different numbers of 4×4 blocks, e.g. a PB of size 64×64 has 256 4×4 blocks and a PB of size 8×4 has only two 4×4 blocks.

The schemes proposed for complexity reduction of uni-prediction, and the grid, raster and the refinement search stages are first implemented individually and then combined in the HM-16.18 encoder. The performance in terms of the different measures is shown for each of these methods and the combined method. In this paper, BD-PSNR and BD-rate are computed for the luminance component.

The Cauchy parameter is set to 2 as given in [32], and the probability constant is set to 99.2% [15]. The values of k_1, k_2, k_3, k_4 are set to 1 and a is set to 8. From Table 2, it is found that the proposed ASR algorithm for uni-prediction has reduced the number of search points, number of 4×4 blocks, motion estimation time and encoding time by 37.87%, 46.15%, 37.28% and 5.78%, respectively. When compared with ESR [17] with the parameters k_1, k_2, k_3, k_4 scaled by 1.4, the proposed algorithm provides nearly the same BD-rate on an average with reduced motion estimation complexity. In our previous work [17], we did not use the upper depth MVD as the fourth search range predictor. Table 3 reports results obtained by using the ESR algorithm [17] as well as ESR when the upper depth MVD is used as the fourth search range predictor, both implemented in HM-16.18 in the LD-P main profile. We observe that the ASR algorithm is better than ESR using upper depth MVD as the fourth search range predictor. In [17] ESR is implemented in HM-16.6 with upper depth PMV disabled in the predicted motion vector stage of TZ search. So the complexity reduction is high. In the current work, upper depth PMV is enabled by default in the beginning of the TZ search algorithm.

Table 2 Performance of the proposed ASR and ESR [17] with parameters scaled by 1.4 implemented in HM-16.18 in LD-P main profile

Class	Sequence	Proposed ASR						ESR with parameters scaled by 1.4					
		BD-rate (%)	BD-PSNR (dB)	ΔN (%)	ΔMET (%)	ΔC (%)	ΔET (%)	BD-rate (dB)	BD-PSNR (%)	ΔN (%)	ΔMET (%)	ΔC (%)	ΔET (%)
B	Kimono	0.07	0.001	48.84	49.23	58.58	8.92	0.03	−0.003	53.20	52.80	61.64	9.86
	ParkScene	−0.07	0.002	32.66	32.72	45.62	3.35	−0.16	−0.000	31.65	31.60	44.38	3.19
	Cactus	0.20	−0.003	43.62	44.01	54.31	5.70	0.09	−0.002	41.17	40.55	50.22	5.31
	BasketballDrive	0.08	0.002	40.70	41.53	49.79	7.96	0.17	−0.001	44.28	44.68	52.81	8.56
	BQTerrace	0.09	−0.003	28.28	28.17	42.02	2.35	0.10	0.000	27.85	27.79	41.81	2.46
C	BasketballDrill	−0.02	−0.004	40.04	40.98	49.25	6.75	0.27	−0.002	40.51	36.14	47.46	5.10
	BQMall	−0.06	0.008	40.77	41.65	49.02	7.18	−0.13	0.005	39.28	35.55	45.00	6.16
	PartyScene	0.06	−0.007	40.34	39.06	45.42	5.43	0.10	−0.007	37.07	32.56	39.90	4.32
	RaceHorsesC	−0.11	0.002	44.85	45.18	51.23	9.13	−0.24	0.014	45.94	43.65	50.37	8.06
D	BasketballPass	0.21	0.001	40.98	41.38	51.63	6.08	0.36	−0.015	39.93	39.69	49.13	6.06
	BQSquare	0.17	−0.014	7.85	2.19	9.11	1.83	0.14	−0.023	6.57	0.76	7.49	1.68
	BlowingBubbles	−0.03	−0.001	37.29	33.94	42.89	3.67	−0.22	0.005	34.30	30.34	37.85	3.53
	RaceHorses	−0.15	−0.005	46.06	44.61	51.11	6.84	−0.11	−0.003	45.69	42.80	47.62	6.13
	Average	0.034	−0.0015	37.87	37.28	46.15	5.78	0.032	−0.0024	37.50	35.30	44.28	5.34

Table 3 Performance of ESR [17] and ESR + upper depth MVD implemented in HM-16.18 in LD-P main profile

Class	Sequence	ESR						ESR + upper depth MVD					
		BD-rate (%)	BD-PSNR (dB)	ΔN (%)	ΔMET (%)	ΔC (%)	ΔET (%)	BD-rate (dB)	BD-PSNR (%)	ΔN (%)	ΔMET (%)	ΔC (%)	ΔET (%)
B	Kimono	0.13	−0.004	54.75	54.99	64.25	10.02	0.14	−0.002	49.11	44.52	49.17	8.15
	ParkScene	0.13	−0.005	33.67	34.34	47.33	3.67	−0.05	0.002	32.79	29.19	37.79	2.91
	Cactus	0.21	−0.003	44.11	44.71	55.02	5.82	−0.02	−0.001	42.05	39.78	47.09	5.25
	BasketballDrive	0.22	−0.005	46.86	48.07	56.97	9.41	0.20	−0.005	40.58	37.71	42.70	7.18
	BQTerrace	0.11	−0.003	28.86	29.19	42.99	2.59	0.14	−0.005	27.79	23.66	31.95	2.17
C	BasketballDrill	0.36	−0.015	43.76	45.02	52.74	8.04	0.28	0.003	40.06	38.65	44.28	6.11
	BQMall	−0.00	0.000	43.97	45.44	51.97	8.68	0.16	−0.008	40.90	39.26	44.34	6.49
	PartyScene	−0.04	−0.006	42.11	41.22	47.08	6.21	0.09	−0.012	40.26	37.13	41.78	4.55
	RaceHorsesC	−0.08	0.005	49.82	50.62	56.48	10.49	−0.25	0.007	45.05	41.52	45.29	7.09
D	BasketballPass	0.35	−0.014	43.66	44.32	54.66	6.60	0.29	−0.012	41.01	38.03	44.76	5.43
	BQSquare	0.17	−0.014	7.85	2.27	9.11	1.81	0.17	−0.014	7.83	2.07	8.41	1.47
	BlowingBubbles	−0.32	0.003	38.21	35.25	43.78	3.88	−0.24	0.002	36.52	31.62	38.30	3.45
	RaceHorses	−0.11	0.004	50.27	48.70	54.93	8.49	0.17	0.005	46.13	42.17	46.12	6.83
	Average	0.086	−0.0043	40.61	40.32	49.02	6.59	0.083	−0.0031	37.70	34.26	40.15	5.16

Table 4 Performance of the proposed grid search complexity reduction compared to the HM-16.18 encoder in the LD-P main profile

Class	Sequence	BD-rate (%)	BD-PSNR (dB)	ΔN(%)	ΔMET(%)	ΔC(%)	ΔET(%)
B	Kimono	0.15	−0.005	39.31	29.28	26.12	5.17
	ParkScene	0.10	−0.005	52.69	40.53	41.51	3.73
	Cactus	0.15	0.001	47.60	37.05	36.85	4.75
	BasketballDrive	0.10	−0.003	33.30	24.87	23.35	4.60
	BQTerrace	0.10	−0.001	58.44	45.36	48.46	3.64
C	BasketballDrill	0.10	−0.003	43.22	35.95	34.46	7.04
	BQMall	0.17	0.008	45.85	38.77	36.90	7.67
	PartyScene	0.23	−0.006	42.22	33.78	33.86	5.56
	RaceHorsesC	0.16	−0.005	31.56	25.20	21.61	5.75
D	BasketballPass	0.50	−0.030	44.91	34.19	33.40	4.49
	BQSquare	-0.05	−0.007	63.50	47.62	56.53	3.79
	BlowingBubbles	0.24	−0.010	45.46	34.06	35.64	3.46
	RaceHorses	0.12	−0.010	34.67	26.16	24.42	4.26
	Average	0.16	−0.006	44.82	34.83	34.86	4.92

Table 5 Performance of the proposed raster search complexity reduction compared to the HM-16.18 encoder in the LD-P main profile

Class	Sequence	BD-rate (%)	BD-PSNR (dB)	ΔN(%)	ΔMET(%)	ΔC(%)	ΔET(%)
B	Kimono	0.10	−0.003	31.07	30.32	38.72	5.39
	ParkScene	−0.13	0.003	17.96	17.90	26.67	1.92
	Cactus	−0.02	0.001	26.31	25.79	35.81	3.25
	BasketballDrive	−0.18	0.003	31.72	31.72	41.13	5.95
	BQTerrace	−0.12	0.003	15.54	13.75	25.37	0.95
C	BasketballDrill	−0.12	0.006	27.63	28.80	36.35	4.43
	BQMall	−0.02	0.008	29.44	30.97	38.44	4.81
	PartyScene	0.22	−0.008	28.23	26.16	34.34	3.11
	RaceHorsesC	0.10	0.001	34.64	34.95	42.10	6.63
D	BasketballPass	0.22	−0.001	27.48	27.06	36.24	3.77
	BQSquare	0.12	−0.005	4.01	2.26	4.23	0.13
	BlowingBubbles	−0.50	−0.000	23.13	18.92	28.53	1.92
	RaceHorses	−0.12	−0.002	34.17	31.11	40.55	4.58
	Average	−0.03	0.000	25.49	24.59	32.96	3.60

From Table 4, it is found that the grid search complexity reduction scheme has reduced the number of search points, motion estimation time, number of 4×4 blocks and encoding time by 44.82%, 34.83%, 34.86% and 4.92% on an average. From Table 5, it is observed that the raster search complexity reduction scheme has reduced these number of search points, motion estimation time, number of 4×4 blocks and encoding time by 25.49%, 24.59%, 32.96% and 3.60% on an average. In Table 6, refinement search complexity reduction has reduced the number of search points, motion estimation time, number of 4×4 blocks and encoding time by 12.73%, 12.20%, 16.28% and 1.84% on an average.

Table 7 shows the results of MCBFME and the proposed scheme implemented in HM-16.18 with AMP SPEED-UP

Table 6 Performance of the proposed refinement search complexity reduction compared to the HM-16.18 encoder in the LD-P main profile

Class	Sequence	BD-rate (%)	BD-PSNR (dB)	ΔN(%)	ΔMET(%)	ΔC(%)	ΔET(%)
B	Kimono	−0.17	0.001	16.45	16.39	19.65	2.93
	ParkScene	0.04	−0.008	10.82	9.99	16.11	0.79
	Cactus	0.03	0.003	11.67	11.52	16.18	1.49
	BasketballDrive	0.05	−0.001	11.39	12.04	16.20	2.18
	BQTerrace	0.11	−0.002	9.51	7.01	13.21	0.55
C	BasketballDrill	0.08	0.007	12.51	14.29	17.09	2.47
	BQMall	0.12	0.002	13.45	15.52	17.38	2.75
	PartyScene	0.08	−0.003	13.56	12.53	16.19	1.83
	RaceHorsesC	−0.24	0.004	14.71	16.23	18.11	3.25
D	BasketballPass	0.04	−0.005	14.31	13.55	18.06	1.95
	BQSquare	0.14	−0.000	4.93	2.63	5.71	0.15
	BlowingBubbles	−0.10	−0.003	14.62	10.87	17.64	1.02
	RaceHorses	−0.14	0.004	17.55	15.99	20.08	2.53
	Average	0.00	−0.000	12.73	12.20	16.28	1.84

enabled by default is shown. Reduction in motion estimation time, number of search points and the number of 4 × 4 for MCBFME and the proposed scheme are 41.65%, 62.57% and 68.20% and 68.14% 77.10% and 77.16% on an average. In Table 8, the results of BMVP and the proposed scheme implemented in HM-16.18 with AMP SPEED-UP disabled are shown. The average reduction in motion estimation time, number of search points and the number of 4 × 4 blocks for BMVP and the proposed scheme are 48.69%, 39.33% and 66.02% and 66.96%, 76.64% and 75.23%. Encoding time reduction of the proposed scheme in HM-16.18 with AMP SPEED-UP enabled and disabled is 9.64% and 8.99%, respectively.

5 Conclusion

The proposed scheme leads to an appreciable reduction in the number of 4 × 4 blocks and motion estimation time without greatly affecting the coding efficiency. Experimental results clearly demonstrate that the proposed grid, raster and refinement search complexity reduction schemes reduce the motion estimation complexity significantly at the respective stages without significantly affecting the video quality and the bitrate. The schemes proposed in this paper are different from existing work as: (1) the grid search MV distance is used for search range reduction in raster search, (2) both grid search MV distance and raster search MV distance are used for search range reduction in refinement search, (3) search points that are already covered in grid search are skipped in refinement search when the grid search MV distance is in between one or two, (4) the early termination criteria proposed for grid and refinement search terminate the stages earlier than in TZ search used in the HM encoder, and (5) a mechanism is proposed for reducing the number of search points in raster search.

Table 7 Performance of MCBFME [27] and the proposed scheme compared to HM-16.18 in LD-P main profile

Class	Sequence	MCBFME [27] implemented in HM-16.18						Proposed scheme implemented in HM-16.18					
		BD-rate (%)	BD-PSNR (%)	ΔN (%)	ΔMET (%)	ΔC (%)	ΔET	BD-rate (%)	BD-PSNR (%)	ΔN (%)	ΔMET (%)	ΔC (%)	ΔET (%)
B	Kimono	−0.07	0.004	70.99	56.09	76.58	9.77	0.06	−0.003	79.24	72.82	79.69	13.33
	ParkScene	0.04	−0.001	50.96	26.69	63.11	2.23	0.20	−0.003	75.91	66.21	77.53	6.57
	Cactus	0.12	−0.003	66.69	48.82	72.37	6.47	0.30	−0.003	79.73	71.75	80.63	9.31
	BasketballDrive	0.32	−0.006	72.59	60.55	79.50	11.46	0.21	−0.004	77.95	72.17	78.83	13.70
	BQTerrace	−0.11	0.002	48.20	20.80	60.95	1.72	0.26	0.000	75.86	64.86	78.18	5.33
C	BasketballDrill	0.61	−0.019	66.68	50.25	71.56	8.68	0.26	−0.014	78.47	71.23	78.81	11.47
	BQMall	0.06	0.003	69.64	54.14	74.90	9.61	0.46	−0.011	79.48	72.84	80.43	12.30
	PartyScene	0.02	−0.005	62.79	41.02	66.96	5.70	0.50	−0.022	76.83	67.55	76.39	8.46
	RaceHorsesC	−0.01	0.001	73.80	60.47	77.58	11.31	0.48	−0.015	77.41	71.63	77.40	14.19
D	BasketballPass	−0.10	0.009	66.13	47.65	73.34	6.90	0.68	−0.022	77.97	69.88	78.71	9.98
	BQSquare	0.17	−0.006	34.14	21.62	31.36	1.03	0.01	−0.000	70.96	51.55	66.07	2.69
	BlowingBubbles	0.15	−0.006	58.63	31.59	62.86	3.47	0.34	−0.004	75.52	63.85	73.94	6.87
	RaceHorses	0.08	−0.002	72.12	55.28	75.52	8.94	0.18	0.005	76.94	69.51	76.45	11.16
	Average	0.10	−0.002	62.57	41.65	68.20	6.72	0.30	−0.007	77.10	68.14	77.16	9.64

Table 8 Performance of BMVP [28] and the proposed scheme compared to HM-16.18 with AMP SPEED-UP disabled in LD-P main profile

Class	Sequence	BMVP [28] implemented in HM-16.18 with AMP SPEED-UP disabled						Proposed scheme implemented in HM-16.18 with AMP SPEED-UP disabled					
		BD-rate (%)	BD-PSNR (%)	ΔN (%)	Δ MET (%)	ΔC (%)	Δ ET (%)	BD-rate (%)	BD-PSNR (%)	ΔN (%)	Δ MET (%)	ΔC (%)	Δ ET (%)
B	Kimono	0.21	−0.006	43.86	54.88	71.56	10.19	0.43	0.008	78.47	71.28	77.05	13.23
	ParkScene	0.38	−0.014	33.14	41.18	59.37	4.21	0.09	−0.004	75.50	65.88	75.67	6.73
	Cactus	0.32	−0.009	42.93	54.17	71.43	6.90	0.27	−0.012	79.40	71.21	78.97	9.07
	BasketballDrive	0.18	−0.002	48.51	61.20	77.23	12.02	−0.13	0.004	77.49	71.04	76.42	13.95
	BQTerrace	−0.14	0.002	31.46	38.03	56.14	3.12	0.16	−0.006	75.58	64.52	75.95	5.29
C	BasketballDrill	0.85	−0.032	43.01	54.78	72.28	7.44	0.22	−0.029	78.07	69.26	76.86	9.41
	BQMall	0.18	−0.008	43.04	54.41	71.57	7.39	0.32	−0.011	79.18	70.86	79.04	9.62
	PartyScene	0.33	−0.012	38.19	47.07	64.81	4.79	0.21	−0.013	76.45	65.74	74.76	6.69
	RaceHorsesC	0.60	−0.019	45.89	58.79	75.54	11.23	0.48	−0.020	76.80	69.03	75.11	13.19
D	BasketballPass	0.29	−0.018	41.15	51.68	69.49	6.91	0.10	−0.001	77.48	68.90	76.65	9.21
	BQSquare	0.12	−0.006	23.75	24.54	41.57	1.31	0.09	−0.010	70.55	51.27	65.03	2.74
	BlowingBubbles	0.09	−0.012	34.20	39.30	56.72	3.92	0.58	−0.018	74.97	63.01	72.08	6.27
	RaceHorses	0.59	−0.024	42.14	52.91	70.55	8.85	0.43	−0.010	76.36	68.44	74.37	11.45
	Average	0.31	−0.012	39.33	48.69	66.02	6.79	0.25	−0.009	76.64	66.96	75.23	8.99

References

1. Sullivan, G.J., Ohm, J.R., Han, W.J., Wiegand, T.: Overview of high efficiency video coding (HEVC) Standard. IEEE Transactions on Circuits and Systems for Video Technology **22**(12), 1648–1667 (2012)
2. Wiegand, T., Sullivan, G.J., Bjontegaard, G., Luthra, A.: Overview of the H.264/AVC video coding standard. IEEE Transactions on Circuits and Systems for Video Technology **13**(7), 560–576 (2003)
3. Nalluri, P., Alves, L.N., Navarro, A.: Fast motion estimation algorithm for HEVC. IEEE Second International Conference on Consumer Electronics (ICCE), Berlin (2012)
4. HEVC test model. https://hevc.hhi.fraunhofer.de/trac/hevc (2018)
5. Jain, J., Jain, A.: Displacement measurement and its application in interframe image coding. IEEE Transactions on Communications. **29**(12), 1799–1808 (1981)
6. Ghanbari, M.: The cross-search algorithm for motion estimation. IEEE Transactions on Communications. **38**(7), 950–953 (1990)
7. Koga, T., Iinuma, K., Hirano, A., Iijima, Y., Ishiguro, T.: Motion compensated interframe coding for video-conferencing. In: Proceedings, National Telecommunications Conference, pp. G5.3.1-G5.3.5 (1981)
8. Po, L.M., Ma, W.C.: A novel four-step search algorithm for fast block motion estimation. IEEE Transactions on Circuits and Systems for Video Technology **6**(3), 313–317 (1996)
9. Zhu, S., Ma, K.K.: A new diamond search algorithm for fast-block matching motion estimation. IEEE Transactions on Image Processing **9**(2), 287–290 (2000)
10. Zhu, C., Lin, X., Chau, L.P.: Hexagon-based search pattern for fast block motion estimation. IEEE Transactions on Circuits and Systems for Video Technology **12**(5), 349–355 (2002)
11. Ismail, Y., McNeely, J.B., Shaaban, M., Mahmoud, H., Bayoumi, M.A.: Fast motion estimation system using dynamic models for H.264/AVC video coding. IEEE Transactions on Circuits and Systems for Video Technology **22**(1), 28–42 (2012)
12. Du, L., Liu, Z., Ikenaga, T., Wang, D.: Linear adaptive search range model for uni-prediction and motion analysis for bi-prediction in HEVC. In: IEEE International Conference on Image Processing (ICIP), pp. 3671-3675 (2014)
13. Nalluri, P., Alves, L.N., Navarro, A.: Complexity reduction methods for fast motion estimation in HEVC. Signal Processing: Image Communication **39**, 280–292 (2015)
14. Ko, Y.H., Kang, H.S., Lee, S.W.: Adaptive search range motion estimation using neighboring motion vector differences. IEEE Transactions on Consumer Electronics **57**(2), 726–730 (2011)
15. Dai, W., Au, O.C., Li, S., Sun, L., Zou, R.: Adaptive search range algorithm based on Cauchy distribution. In: IEEE Visual Communications and Image Processing (VCIP), pp. 1-5 (2012)
16. Lee, T.K., Chan, Y.L., Siu, W.C.: Adaptive search range for HEVC motion estimation based on depth information. IEEE Transactions on Circuits and Systems for Video Technology **27**(10), 2216–2230 (2017)
17. Varma, K.C.R.C., Kumar, M.V.P., Mahapatra, S.: Search range reduction for uni-prediction and bi-prediction in HEVC. J. Real-Time Image Proc. **16**, 1351–1364 (2019)
18. Ismail, Y., McNeely, J.B., Shaaban, M., Bayoumi, M.A.: A generalized fast motion estimation algorithm using external and internal stop search techniques for H.264 video coding standard. In: IEEE International Symposium on Circuits and Systems, (ISCAS), pp. 3574–3577. IEEE, Seattle, WA (2008)
19. Liu, B., Zaccarin, A.: New fast algorithms for the estimation of block motion vectors. IEEE Transactions on Circuits and Systems for Video Technology **3**(2), 148–157 (2012)
20. Cheung, C.K., Po, L.M.: Normalized partial distortion search algorithm for block motion estimation. IEEE Transactions on Circuits and Systems for Video Technology **10**(3), 417–422 (2000)
21. Cheung, C.H., Po, L.M.: Adjustable Partial Distortion Search Algorithm for Fast Block Motion Estimation. IEEE Transactions on Circuits and Systems for Video Technology **13**(1), 100–110 (2003)
22. Varma, K.C.R.C., Kumar, M.V.P., Mahapatra, S.: A low complexity block matching algorithm for fast motion estimation in High Efficiency Video Coding. In: IEEE National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), pp. 1-4 (2015)
23. Jeong, J.H., Parmar, N., Sunwoo, M.H.: Enhanced test zone search algorithm with rotating pentagon search. In: IEEE International SoC Design Conference (ISOCC), pp. 275-276 (2015)
24. Gao, L., Dong, S., Wang, W., Wang, R., Gao, W.: A novel integer-pixel motion estimation algorithm based on quadratic prediction. In: IEEE International Conference on Image Processing (ICIP), pp. 2810-2814 (2015)
25. Kim, S., Park, C., Chun, H., Kim, J.: A novel fast and low-complexity motion estimation for UHD HEVC. Picture Coding Symposium (PCS), IEEE, pp. 105-108 (2013)
26. Chen, Y.W., Hsiao, M.H., Chen, H.T., Liu, C.Y., Lee, S.Y.: Content-aware fast motion estimation algorithm. Journal of Visual Communication and Image Representation **19**(4), 256–269 (2008)
27. Fan, R., Zhang, Y., Li, B.: Motion Classification-Based Fast Motion Estimation for High-Efficiency Video Coding. IEEE Transactions on Multimedia **19**(5), 893–897 (2017)
28. Kim, T.S., Rhee, C.E., Lee, H.J., Chae, S.I.: Fast integer motion estimation with bottom-up motion vector prediction for an HEVC encoder. IEEE Trans. Circuits Syst. Video Technol. **28**(12), 3398–3411 (2017)
29. Bossen, F., Common, H.: Test conditions and software reference configurations. JCTVC-L1100 (2013)
30. Test sequences. ftp://ftp.tnt.uni-hannover.de (2012)
31. Bjontegaard, G.: Improvements of the BD-PSNR model. ITU-T SG16, VCEG-AI11, July (2008)
32. Kamaci, N., Altunbasak, Y., Mersereau, R.M.: Frame bit allocation for the H.264/AVC video coder via Cauchy-density based rate and distortion models. IEEE Transactions on Circuits and Systems for Video Technology **15**(8), 994–1006 (2005)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

K C Ravi Chandra Varma received B.Tech. degree in Electronics and Communication Engineering from Swami Vivekananda Institute of Technology, India in 2010 and M.Tech. degree in Electronics and Communication engineering from NIT Calicut, India in 2012. He is currently pursuing Ph.D. degree in Electronics and Electrical Communication Engineering from IIT Kharagpur, India. His current research includes video coding algorithms and multiview video coding.

Sudipta Mahapatra graduated in Electronics and Telecommunication Engineering from University College of Engineering, Burla, Sambalpur University, Orissa, India in the year 1990. He obtained his M.Tech. and Ph.D. degrees in Computer Engineering from IIT Kharagpur in the years 1992 and 1997, respectively. From April 1993 to September 2002, he was working in the Computer Science and Engineering department of National Institute of Technology, Rourkela. He was in the Electronic Systems Design Group of Loughborough University, UK, as a BOYSCAST Fellow of DST, Government of India, from March 1999 to March 2000. He joined the Electronics and Electrical Communication Engineering Department of IIT Kharagpur in September 2002 where currently he is working as an Associate Professor. His areas of research interest include: Video coding/streaming, Optical and Wireless Networking, and Parallel and Distributed Systems.